

# A Deep Cascade of Convolutional Neural Networks for Dynamic MR Image Reconstruction

Jo Schlemper\*, Jose Caballero, Joseph V. Hajnal, Anthony Price and Daniel Rueckert, *Fellow, IEEE*

**Abstract**—Inspired by recent advances in deep learning, we propose a framework for reconstructing dynamic sequences of 2D cardiac magnetic resonance (MR) images from undersampled data using a deep cascade of convolutional neural networks (CNNs) to accelerate the data acquisition process. In particular, we address the case where data is acquired using aggressive Cartesian undersampling. Firstly, we show that when each 2D image frame is reconstructed independently, the proposed method outperforms state-of-the-art 2D compressed sensing approaches such as dictionary learning-based MR image reconstruction, in terms of reconstruction error and reconstruction speed. Secondly, when reconstructing the frames of the sequences jointly, we demonstrate that CNNs can learn spatio-temporal correlations efficiently by combining convolution and data sharing approaches. We show that the proposed method consistently outperforms Dictionary Learning with Temporal Gradients (DLTG) and is capable of preserving anatomical structure more faithfully up to 11-fold undersampling. Moreover, reconstruction is very fast: each complete dynamic sequence can be reconstructed in less than 10s and, for the 2D case, each image frame can be reconstructed in 23ms, enabling real-time applications.

**Index Terms**—Deep learning, convolutional neural network, dynamic magnetic resonance imaging, compressed sensing, image reconstruction.

## I. INTRODUCTION

IN many clinical scenarios, medical imaging is an indispensable diagnostic and research tool. One such important modality is Magnetic Resonance Imaging (MRI), which is non-invasive and offers excellent resolution with various contrast mechanisms to reveal different properties of the underlying anatomy. However, MRI is associated with an inherently slow acquisition process. This is because data samples of an MR image are acquired sequentially in  $k$ -space and the speed at which  $k$ -space can be traversed is limited by physiological and hardware constraints [1]. A long data acquisition procedure imposes significant demands on patients, making this imaging modality expensive and less accessible. One possible approach to accelerate the acquisition process is to undersample  $k$ -space, which in theory provides an acceleration rate proportional to a reduction factor of a number of  $k$ -space traversals required. However, undersampling in  $k$ -space violates the Nyquist-Shannon theorem and

generates aliasing artefacts when the image is reconstructed. The main challenge in this case is to find an algorithm that can recover an uncorrupted image taking into account the undersampling regime combined with a-priori knowledge of appropriate properties of the image to be reconstructed.

Using Compressed Sensing (CS), images can be reconstructed from sub-Nyquist sampling, assuming the following: firstly, the images must be *compressible*, i.e. they have a sparse representation in some transform domain. Secondly, one must ensure *incoherence* between the sampling and sparsity domains to guarantee that the reconstruction problem has a unique solution and that this solution is attainable. In practice, this can be achieved with random subsampling of  $k$ -space, which produces aliasing patterns in the image domain that can be regarded as correlated noise. Under such assumptions, images can then be reconstructed through nonlinear optimisation or iterative algorithms. The class of methods which apply CS to the MR reconstruction problem is termed CS-MRI [1]. In general, these methods use a fixed sparsifying transforms, e.g. wavelet transformations. A natural extension of these approaches has been to enable more flexible representations with *adaptive* sparse modelling, where one attempts to learn the optimal sparse representation from the data directly. This can be done by exploiting, for example, dictionary learning (DL) [2].

To achieve more aggressive undersampling, several strategies can be considered. One way is to further exploit the inherent redundancy of the MR data. For example, in dynamic imaging, one can make use of spatio-temporal redundancies [3], [4], [5], or when imaging a full 3D volume, one can exploit redundancy from adjacent slices [6]. An alternative approach is to exploit sources of explicit redundancy of the data to turn the initially underdetermined problem arising from undersampling into a determined or overdetermined problem that is easily solved. This is the fundamental assumption underlying parallel imaging [7]. Similarly, one can make use of multi-contrast information [8] or the redundancy generated by multiple filter responses of the image [9]. These explicit redundancies can also be used to complement the sparse modelling of inherent redundancies [10], [11].

Recently, deep learning has been successful at tackling many computer vision problems. Deep neural network architectures, in particular convolutional neural networks (CNNs), are becoming the state-of-the-art technique for various imaging problems including image classification [12], object localisation [13] and image segmentation [14]. Deep architectures are capable of extracting features from data to build increasingly abstract representations, replacing the traditional

\*J. Schlemper is with the Department of Computing, Imperial College London, SW7 2AZ London, U.K. (e-mail: jo.schlemper11@imperial.ac.uk).

A. N. Price and J. V. Hajnal are with the Division of Imaging Sciences and Biomedical Engineering Department, King's College London, St. Thomas' Hospital, SE1 7EH London, U.K. (email: anthony.price@kcl.ac.uk; jo.hajnal@kcl.ac.uk).

J. Caballero and D. Rueckert is with the Department of Computing, Imperial College London, SW7 2AZ London, U.K. (e-mail: jose.caballero06@imperial.ac.uk; d.rueckert@imperial.ac.uk).

approach of carefully hand-crafting features and algorithms. For example, it has already been demonstrated that CNNs outperform sparsity-based methods in super-resolution [15] in terms of both reconstruction quality and speed [16]. One of the contributions of our work is to explore the application of CNNs in undersampled MR reconstruction and investigate whether they can exploit data redundancy through learned representations. In fact, CNNs have already been applied to compressed sensing from random Gaussian measurements [17]. Despite the popularity of CNNs, there has only been preliminary research on CNN-based MR image reconstruction [18], [19], hence the applicability of CNNs to this problem for various imaging protocols has yet to be fully explored.

In this work we consider reconstructing dynamic sequences of 2D cardiac MR images with Cartesian undersampling, as well as reconstructing each frame independently, using CNNs. We view the reconstruction problem as a de-aliasing problem in the image domain. Reconstruction of undersampled MR images is challenging because the images typically have low signal-to-noise ratio, yet often high-quality reconstructions are needed for clinical applications. To resolve this issue, we propose a deep network architecture which forms a cascade of CNNs. Our cascade network closely resembles the iterative reconstruction of DL-based methods, however, our approach allows end-to-end optimization of the reconstruction algorithm. We compare the CNN method with two state-of-the-art DL methods for MR image reconstruction, namely *DLMRI* [2] for 2D reconstruction and Dictionary Learning with Temporal Gradient (*DLTG*) [3] for dynamic reconstruction in the context of cardiac MRI. *DLMRI* learns a dictionary of spatial features, whereas *DLTG* further exploits the temporal correlation by learning spatio-temporal features and imposing total variation penalty along temporal-axis. We show that the proposed method outperforms them in terms of reconstruction error and perceptual quality, especially for aggressive undersampling rates. Moreover, owing to GPU-accelerated libraries, images can be reconstructed efficiently using the approach. In particular, for 2D reconstruction, each image can be reconstructed in about 23ms, which is fast enough to enable real-time applications. For the dynamic case, sequences can be reconstructed within 10s, which is reasonably fast for off-line reconstruction methods.

## II. PROBLEM FORMULATION

Let  $\mathbf{x} \in \mathbb{C}^N$  represent a sequence of 2D complex-valued MR images stacked as a column vector, where  $N = N_x N_y N_t$ . Our problem is to reconstruct  $\mathbf{x}$  from  $\mathbf{y} \in \mathbb{C}^M$ , the measurements in  $k$ -space, such that:

$$\mathbf{y} = \mathbf{F}_u \mathbf{x} \quad (1)$$

Here  $\mathbf{F}_u \in \mathbb{C}^{M \times N}$  is an undersampled Fourier encoding matrix. For undersampled  $k$ -space measurements ( $M \ll N$ ), the system of equations (1) is underdetermined and hence the inversion process is ill-defined. In order to reconstruct  $\mathbf{x}$ , one must exploit a-priori knowledge of its properties, which can be done by formulating an unconstrained optimisation problem:

$$\min_{\mathbf{x}} \mathcal{R}(\mathbf{x}) + \lambda \|\mathbf{y} - \mathbf{F}_u \mathbf{x}\|_2^2 \quad (2)$$

$\mathcal{R}$  expresses regularisation terms on  $\mathbf{x}$  and  $\lambda$  allows the adjustment of data fidelity based on the noise level of the acquired measurements  $\mathbf{y}$ . For CS-based methods, the regularisation terms  $\mathcal{R}$  typically involve  $\ell_0$  or  $\ell_1$  norms in the sparsifying domain of  $\mathbf{x}$ . Our formulation is inspired by DL-based reconstruction approaches [2], in which the problem is formulated as:

$$\min_{\mathbf{x}, \mathbf{D}, \{\gamma_i\}} \sum_i (\|\mathbf{R}_i \mathbf{x} - \mathbf{D} \gamma_i\|_2^2 + \nu \|\gamma_i\|_0) + \lambda \|\mathbf{y} - \mathbf{F}_u \mathbf{x}\|_2^2 \quad (3)$$

Here  $\mathbf{R}_i$  is an operator which extracts a spatio-temporal image patch at  $i$ ,  $\gamma_i$  is the corresponding sparse code with respect to a dictionary  $\mathbf{D}$ . In this approach, the regularisation terms force  $\mathbf{x}$  to be approximated by the reconstructions from the sparse code of patches. By taking the same approach, for our CNN formulation, we force  $\mathbf{x}$  to be well-approximated by the CNN reconstruction:

$$\min_{\mathbf{x}} \|\mathbf{x} - f_{\text{cnn}}(\mathbf{x}_u | \boldsymbol{\theta})\|_2^2 + \lambda \|\mathbf{F}_u \mathbf{x} - \mathbf{y}\|_2^2 \quad (4)$$

Here  $f_{\text{cnn}}$  is the forward mapping of the CNN parameterised by  $\boldsymbol{\theta}$ , possibly containing millions of adjustable network weights, which takes in the zero-filled reconstruction  $\mathbf{x}_u = \mathbf{F}_u^H \mathbf{y}$  and directly produces a reconstruction as an output. Since  $\mathbf{x}_u$  is heavily affected by aliasing from sub-Nyquist sampling, the CNN reconstruction can therefore be seen as solving a de-aliasing problem in the image domain. The approach of Eq. (4), however, is limited in the sense that the CNN reconstruction and the data fidelity are two independent terms. In particular, since the CNN operates in the image domain, it is trained to reconstruct the sequence without a-priori information of the acquired data in  $k$ -space. However, if we already know some of the  $k$ -space values, then the CNN should be discouraged from modifying them. Therefore, by incorporating the data fidelity in the learning stage, the CNN should be able to achieve better reconstruction. This means that the output of the CNN is now conditioned on  $\Omega$ , an index set indicating which  $k$ -space measurements have been sampled in  $\mathbf{y}$ . Then, our final reconstruction is given simply by the output,  $\mathbf{x}_{\text{cnn}} = f_{\text{cnn}}(\mathbf{x}_u | \boldsymbol{\theta}, \lambda, \Omega)$ . Given training data  $\mathcal{D}$  of input-target pairs  $(\mathbf{x}_u, \mathbf{x}_t)$  where  $\mathbf{x}_t$  is a fully-sampled data, we can train the CNN to produce an output that attempts to accurately reconstruct the data by minimising an objective function:

$$\mathcal{L}(\boldsymbol{\theta}) = \sum_{(\mathbf{x}_u, \mathbf{x}) \in \mathcal{D}} \ell(\mathbf{x}, \mathbf{x}_{\text{cnn}}) \quad (5)$$

where  $\ell$  is a loss function. In this work, we consider an element-wise squared loss, which is given by  $\ell(\mathbf{x}_t, \mathbf{x}_{\text{cnn}}) = \|\mathbf{x}_t - \mathbf{x}_{\text{cnn}}\|_2^2$ .

## III. DATA CONSISTENCY LAYER

Denote a Fourier transform of data  $\mathbf{x}$  as  $\hat{\mathbf{x}} = \mathbf{F} \mathbf{x}$ , where  $\mathbf{F}$  is the Fourier encoding matrix. In order to incorporate the data

fidelity in the network architecture, we first note the following: for a fixed  $\theta$ , Eq. (4) has a closed-form solution  $\hat{\mathbf{x}}_{\text{rec}}$  in  $k$ -space, given as in [2]:

$$\hat{\mathbf{x}}_{\text{rec}}(k) = \begin{cases} \hat{\mathbf{x}}_{\text{cnn}}(k) & \text{if } k \notin \Omega \\ \frac{\hat{\mathbf{x}}_{\text{cnn}}(k) + \lambda \hat{\mathbf{x}}_u(k)}{1 + \lambda} & \text{if } k \in \Omega \end{cases} \quad (6)$$

where  $\hat{\mathbf{x}}_{\text{cnn}} = \mathbf{F}f_{\text{cnn}}(\mathbf{x}_u|\theta)$ . The final reconstruction is obtained by applying the inverse of the encoding matrix  $\mathbf{x}_{\text{rec}} = \mathbf{F}^{-1}\hat{\mathbf{x}}_{\text{rec}}$ . In the limit  $\lambda \rightarrow \infty$  we simply replace the  $k$ th predicted coefficient by the original coefficient if it has been sampled. For this reason, this operation is called a *data consistency step* in  $k$ -space (DC).

Since the DC step has a simple expression, we can in fact treat it as a layer operation of the network, which we denote as a *DC layer*. When defining a layer of a network, the rules for forward and backward passes must be specified in order for the network to be end-to-end trainable. This is because CNN training can effectively be performed through stochastic gradient descent, where one updates the network parameters  $\theta$  to minimise the objective function  $\mathcal{L}$  by descending along the direction given by the derivative  $\partial\mathcal{L}/\partial\theta^T$ . Therefore, it is necessary to define the gradients of each network layer relative to the network's output. In practice, one uses an efficient algorithm called *backpropagation* [20], where the final gradient is given by the product of all the Jacobians of the layers contributing to the output. Hence, in general, it suffices to specify a layer operation  $f_L$  for the forward pass and derive the Jacobian of the layer with respect to the layer input  $\partial f_L/\partial \mathbf{x}^T$  for the backward pass.

*a) Forward pass:* The data consistency in  $k$ -space can be simply decomposed into three operations: Fourier transform, data consistency and inverse Fourier transform. In our case, the applied Fourier transform is a stack of two-dimensional (2D) discrete Fourier transforms (DFT) applied to each image frame of  $\mathbf{x}$ , which is written as  $\hat{\mathbf{x}} = \mathbf{F}\mathbf{x}$  in matrix form. The inverse transformation is defined analogously, where  $\mathbf{x} = \mathbf{F}^{-1}\hat{\mathbf{x}}$ . The data consistency  $f_{dc}$  performs the element-wise operation defined in Eq. (6), which we can write it in matrix form as:

$$f_{dc}(\hat{\mathbf{x}}, \hat{\mathbf{x}}_u; \lambda) = \mathbf{\Lambda}\hat{\mathbf{x}} + \frac{\lambda}{1 + \lambda}\hat{\mathbf{x}}_u \quad (7)$$

Here  $\mathbf{\Lambda}$  is a diagonal matrix of the form:

$$\mathbf{\Lambda}_{kk} = \begin{cases} 1 & \text{if } k \notin \Omega \\ \frac{1}{1 + \lambda} & \text{if } k \in \Omega \end{cases} \quad (8)$$

Combining the three operations defined above, we can obtain the forward pass of the layer performing data consistency in  $k$ -space:

$$f_L(\mathbf{x}, \hat{\mathbf{x}}_u; \lambda) = \mathbf{F}^{-1}\mathbf{\Lambda}\mathbf{F}\mathbf{x} + \frac{\lambda}{1 + \lambda}\mathbf{F}^{-1}\hat{\mathbf{x}}_u \quad (9)$$

*b) Backward pass:* In general, one requires *Wirtinger calculus* to derive a gradient in complex domain [21]. However, in our case, the derivation greatly simplifies due to the linearity of the DFT matrix and the data consistency operation.

The Jacobian of the DC layer with respect to the layer input  $\mathbf{x}$  is therefore given by:

$$\frac{\partial f_L}{\partial \mathbf{x}^T} = \mathbf{F}^{-1}\mathbf{\Lambda}\mathbf{F} \quad (10)$$

Note that unlike many other applications where CNNs process real-valued data, MR images are complex-valued and the network needs to account for this. One possibility would be to design the network to perform complex-valued operations. A simpler approach, however, is to accommodate the complex nature of the data with real-valued operations in a dimensional space twice as large (i.e. we replace  $\mathbb{C}^N$  by  $\mathbb{R}^{2N}$ ). In the latter case, the derivations above still hold due to the fundamental assumption in Wirtinger calculus.

#### IV. CASCADING NETWORK

For CS-based methods, in particular for DL-based methods, the optimisation problem such as in Eq. (3) is solved using a coordinate-descent type algorithm, alternating between the de-aliasing step and the data consistency step until convergence. In contrast, with CNNs, we are performing one step de-aliasing and the same network cannot be used to de-alias iteratively. While CNNs may be powerful enough to learn one step reconstruction, such a network could show signs of overfitting, unless there is vast amounts of training data. In addition, training such networks may require a long time as well as careful fine-tuning steps. It is therefore best to be able to use CNNs for iterative reconstruction approaches.

A simple solution is to train a second CNN which learns to reconstruct from the output of the first CNN. In fact, we can concatenate a new CNN on the output of the previous CNN to build extremely deep networks which iterate between intermediate de-aliasing and the data consistency reconstruction. We term this a *cascading network*. In fact, one can essentially view this as unfolding the optimisation process of DLMRI. If each CNN expresses the dictionary learning reconstruction step, then the cascading CNN can be seen as a direct extension of DLMRI, where the whole reconstruction pipeline can be optimised from training, as seen in Fig. 3.

#### V. DATA SHARING LAYER

For the case of reconstructing dynamic sequences, the temporal correlation between frames can be exploited as an additional regulariser to further de-alias the undersampled images. In addition to using 3D convolution to learn spatio-temporal features of the input sequence, we propose incorporating features that could benefit the CNN reconstruction, inspired by *data sharing* approaches [22], [23], [24]: if the change in image content is relatively small for any adjacent frames, then the neighbouring  $k$ -space samples along the temporal-axis often capture similar information. In fact, as long as this assumption is valid, for each frame, we can fill the entries using the samples from the adjacent frames to approximate missing  $k$ -space samples. Specifically, for each frame  $t$ , all frames from  $t - n_{\text{adj}}$  to  $t + n_{\text{adj}}$  are considered, filling the missing  $k$ -space samples at frame  $t$ . If more than one frame within the range contains a sample, we take the weighted average of the samples. The obtained image can be

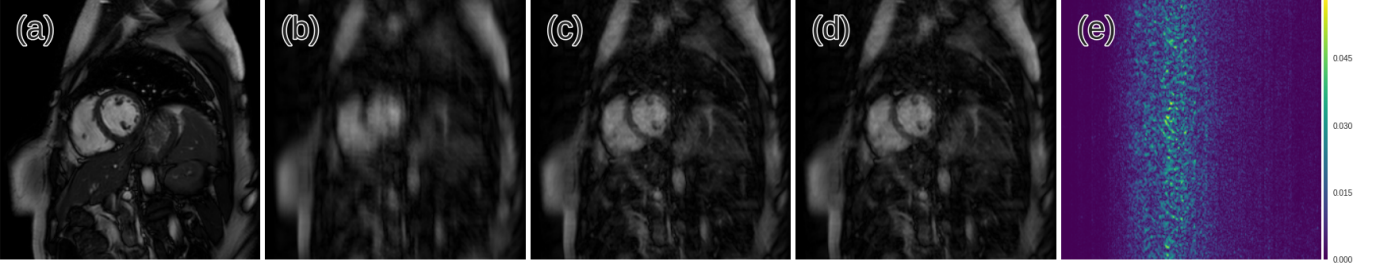


Fig. 1. Illustration of data sharing approach. (a) The ground truth image (b) 12-fold undersampling using the mask in Fig. 2a (c) Image generated by data sharing with  $n_{adj} = 2$ , *simulating* 4-fold undersampling by filling the entries shown in Fig. 2b (d) Actual 4-fold undersampling of data using the mask in Fig. 2b (e) The difference between (c) and (d). One can notice that the images are similar except for the data inconsistency of the dynamic content around heart.

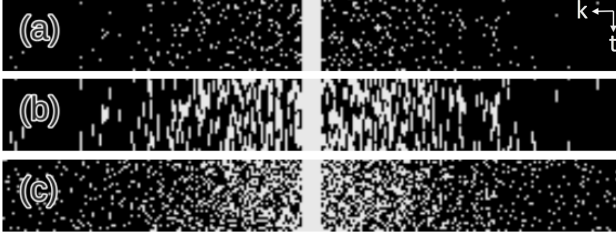


Fig. 2. Cartesian undersampling mask displayed for  $k$ - $t$  axes, where sampled entries are indicated in light gray. The frequency encoding direction of  $k$ -space is fully sampled and hence omitted for simplicity. (a) 12-fold undersampling mask (b) Entries that are filled using data sharing with  $n_{adj} = 2$ , conceptually simulating the 4-fold undersampling mask (c) Random 4-fold undersampling mask. Note that (c) achieves higher incoherence compared to (b) and therefore is preferred for the CS framework.

seen as an image acquired with lower undersampling factor as seen in Fig. 1 using sampling pattern shown in Fig. 2.

In practice, however, cardiac sequences contain highly dynamic content around the heart and hence combining the adjacent frames results in data inconsistency around the dynamic region, as shown in Fig. 1(e). However, for CNN reconstruction, we can incorporate these images as an extra input to train the network rather than directly incorporating it in our final prediction. For our network, we implement *data sharing (DS) layers* which take an input image and generate multiple “data-shared” images for a range of  $n_{adj}$ . The resulting images are concatenated along the channel-axis and treated as a new input fed into the first convolution layer of the CNNs. Therefore, using the images obtained from data sharing can be interpreted as transforming the problem into joint estimation of aliasing as well as the dynamic motion, where the effect of aliasing is considerably smaller. Note that for the cascading network architecture, from the second subnetwork onwards, the input to each subnetwork is no longer “undersampled”, but instead contains intermediate predicted values from the previous subnetwork. In this case, we average all the entries from the adjacent frames and update the samples which were not initially acquired. For this work, we allocate equal weight on all adjacent  $k$ -space samples, however, in future, more elaborate averaging schemes will be considered. We will empirically show the benefit of this approach in the experiment section.

## VI. ARCHITECTURE AND IMPLEMENTATION

Incorporating all the new elements mentioned above, we can devise our cascading network architecture. Our CNN takes in a two-channelled sequence of images  $\mathbb{R}^{2N_x N_y N_t}$ , where the channels store real and imaginary parts of the undersampled images. Based on literature, we used the following network architecture for the CNN, illustrated in Fig. 3: it has  $n_d - 1$  (3D) convolution layers  $C_i$ , which are all followed by Rectifier Linear Units (ReLU) as a choice of nonlinearity. For each of them, we used a kernel size  $k = 3$  [25] and the number of filters was set to  $n_f = 64$ . The final layer of the CNN module is a convolution layer  $C_{rec}$  with  $k = 3$  and  $n_f = 2$ , which projects the extracted representation back to the image domain. We also used *residual connection* [12], which sums the output of the CNN module with its input. Finally, we form a cascading network by using the DC layers interleaved with the CNN reconstruction modules  $n_c$  times. For DS layer, we take the input to each subnetwork, generating images for all  $n_{adj} \in \{0, 1, \dots, 5\}$ . As aforementioned, the resulting images are concatenated along the channel-axis and fed to the first convolution layer. We found that this choice of architecture works sufficiently well, however, the parameters were not optimised and there is therefore room for refinement of the results presented. Hence the result is likely to be improved by, for example, incorporating pooling layers and varying the parameters such as kernel size and stride [14], [26].

As mentioned, pixel-wise squared error was used as the objective function. The minibatch size was set to 10, however, for the deeper models with large number of cascades, the minibatch size was reduced to fit the model on a single GPU memory. We initialised the network weights using He initialisation [27]. The Adam optimiser [28] was used to train all models, with parameters  $\alpha = 10^{-4}$ ,  $\beta_1 = 0.9$  and  $\beta_2 = 0.999$  unless specified. We also added  $\ell_2$  weight decay of  $10^{-7}$ .

Our model can also be used for 2D image reconstruction by setting  $N_t = 1$  and use 2D convolution layers instead. Note also that data sharing does not apply to 2D models. For the following experiments, we first explore the network configurations by considering 2D MR image reconstruction. We identify our network by the values of  $n_c$ ,  $n_d$  and the use of data sharing. For example, *D5-C2* means a network with  $n_d = 5$ ,  $n_c = 2$  with no data sharing. *D5-C10(S)* corresponds

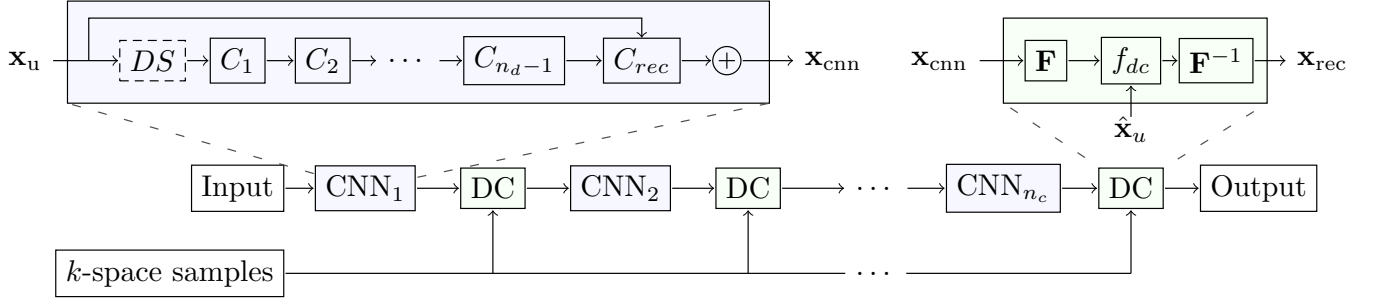


Fig. 3. A cascade of CNNs. DC denotes the data consistency layer and DS denotes the data sharing layer. The number of convolution layers within each network and the depth of cascade is denoted by  $n_d$  and  $n_c$  respectively. Note also that DS layer only applies when the input is a sequence of images.

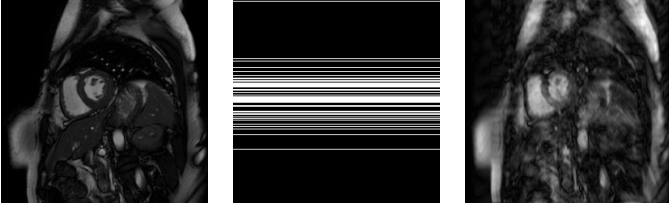


Fig. 4. (Left) Ground truth (Middle) Cartesian undersampling mask for 4-fold acceleration and (Right) A zero-filled reconstruction.

a network with  $n_d = 5$ ,  $n_c = 10$  and data sharing.

## VII. EXPERIMENTAL RESULTS

### A. Setup

*a) Dataset:* Our method was evaluated using the cardiac MR dataset used in [3], consisting of 10 fully sampled short-axis cardiac cine MR scans. Each scan contains a single slice SSFP acquisition with 30 temporal frames with a  $320 \times 320$  mm field of view and 10 mm slice thickness. The raw data consists of 32-channel data with sampling matrix size  $192 \times 190$ , which was zero-filled to the matrix size  $256 \times 256$ . The data was reconstructed using SENSE [29] with no undersampling and retrospective gating. Coil sensitivity maps were normalized to a body coil image and used to produce a single complex-valued reconstructed image. For the following experiments, the complex valued images were back-transformed to regenerate  $k$ -space samples, simulating a fully sampled single-coil acquisition. The  $k$ -spaces were then retrospectively undersampled using Cartesian undersampling masks adopted from [4]: for each frame we acquired the eight lowest spatial frequencies. The sampling probability of other entries along the phase-encoding direction was determined by a zero-mean Gaussian distribution. The acceleration rates are stated with respect to the matrix size of the raw data.

*b) Metric:* We used mean squared error (MSE) as our quantitative measure. The reconstruction signal-to-noise ratio from undersampled data is highly dependent on the imaging data and the undersampling mask. To take this into consideration for fair comparison, we assigned an arbitrary but fixed undersampling mask for each image in the test data, yielding a fixed number of image-mask pairs to be evaluated.

### B. Reconstruction of 2D Images

For the following experiments, we split the dataset into training and testing sets including five subjects each. Each image frame was then treated as an independent image, yielding a total of 150 images per set. Unless specified, the undersampling rate was set to 3-fold. The undersampling masks were generated on-the-fly to allow the network to learn diverse patterns of aliasing artefact. Note that validation set was not utilised due to the limited number of data. Instead, we let the network train up to a fixed number of backpropagations. We also applied rigid transformation including shift, flip and rotation for data augmentation to train the network on diverse range of input patterns.

*1) Trade-offs between  $n_d$  and  $n_c$ :* In this experiment we compared two architectures: *D5-C2* ( $n_d = 5, n_c = 2$ ) and *D11-C1* ( $n_d = 11, n_c = 1$ ) to evaluate the benefit of the DC step. The two networks have equivalent depths when the DC layers are viewed as feature extraction layers. However, the former can build deeper features of the image, whereas the latter benefits from the intermediate data consistency step. Each network was trained end-to-end for  $3 \times 10^5$  backpropagations and MSE on the training and test data is shown in Fig. 5. One can observe that *D11-C1* quickly overfitted the training data. On the other hand, both train and test errors for *D5-C2* were notably lower. Since our dataset is small, deep networks can overfit more easily. However, despite the depth, *D5-C2* did not show overfitting. This is because the architecture employs two data consistency steps and rebuilds the representations at each cascading iteration. This suggests that it is more beneficial to interleave DC layers projecting the acquired  $k$ -space onto intermediate reconstructions with the CNN image reconstruction modules, which appears to help both the reconstruction as well as the generalisation.

*2) Effect of Cascading Iterations  $n_c$ :* In this experiment, we explored how much benefit the network can get by increasing the cascading iteration. We fixed the architectures to have  $n_d = 5$ , but varied the cascading iteration from  $n_c \in \{1, 2, 3, 4, 5\}$ . For this section, due to time constraints, we trained the networks using a greedy approach: we initialised the cascading net with  $n_c = k$  using the net with  $n_c = k-1$  that was already trained. For each  $n_c$ , we performed up to  $10^5$  backpropagations. Note that the greedy approach leads to a satisfactory solution, however, better results can be



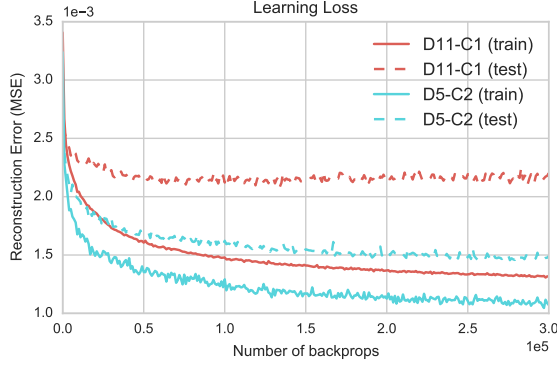


Fig. 5. A comparison of the networks with and without the intermediate DC step. *D5-C2* shows superior performance over *D11-C1*. In particular, *D5-C2* has considerably lower test error, showing an improved generalization property.

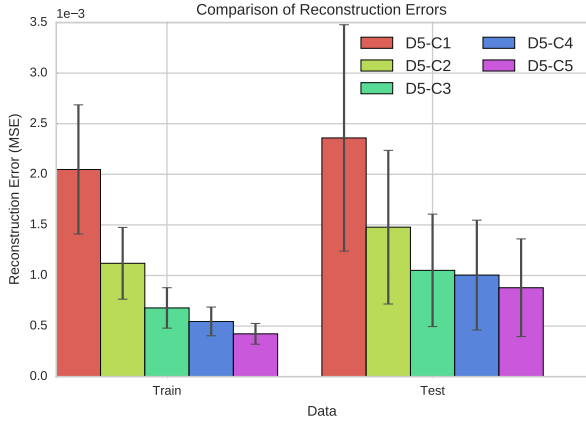


Fig. 6. The effect of increasing cascading iteration  $n_c$ . One can see that the reconstruction error on both training and test data monotonically decreases as  $n_c$  increases. However, the rate of improvement is reduced after  $n_c = 3$ .

achieved with random initialisation, as initialising a network from another networks convergence point can make it more likely that it gets stuck in suboptimal local minima.

Reconstruction errors for the cascading depth tests are shown in Fig. 6. We observed that while deeper cascading nets tend to overfit more, they still reduced the test error every time. The rate of improvement was reduced after 3 cascading layers, however, we see that the standard deviation of error was also reduced for the deeper models. We also visualised the intermediate reconstructions (output of each cascading iteration) within *D5-C5*, shown in Fig. 7. In general, we see that the cascading net gradually recovers and sharpens the output image. Although the reconstruction error decreased monotonically at each cascading depth, we observed that the output of the fourth subnetwork appears to be more grainy than the output of the preceding subnetwork. This suggests the benefit of the end-to-end training scheme: since we are optimising the whole pipeline of reconstruction, the additional CNNs are internally used to rectify the error caused by the previous CNNs. In this case, the fourth subnetwork appears to counteract over-smoothing in the third subnetwork.

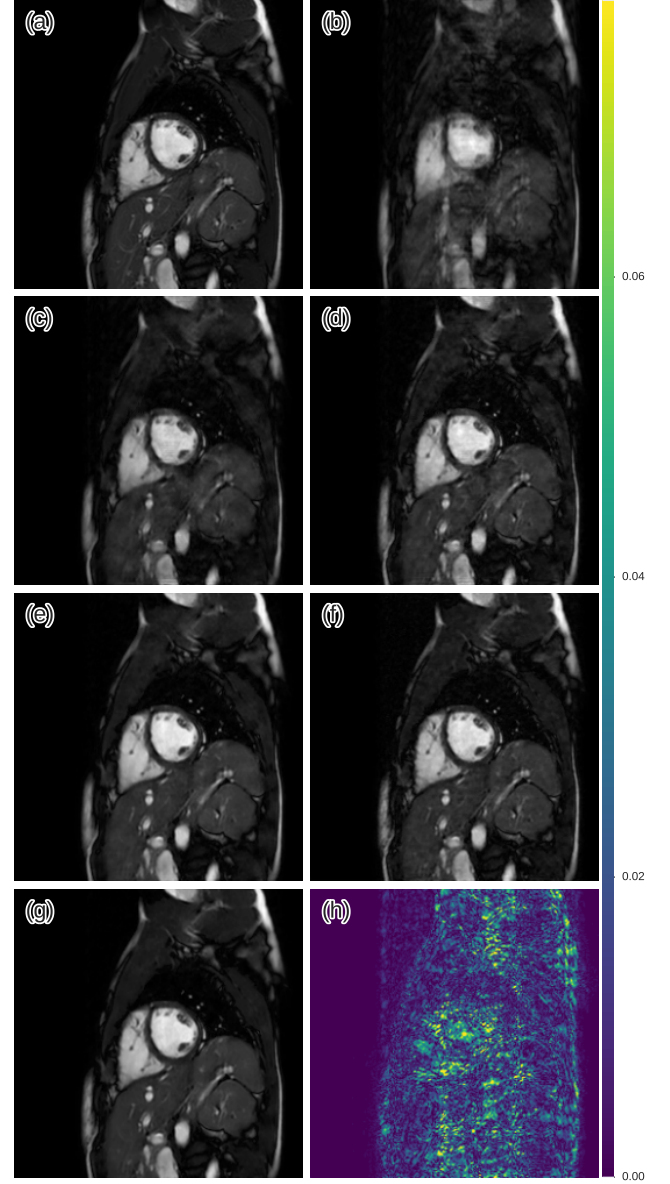


Fig. 7. The inspection of the output of each cascading subnetwork in *D5-C5*. (a) Ground truth (b) The input to the network that was 3x undersampled, The output of (c) first, (d) second, (e) third, (f) fourth cascading subnetwork respectively. (g,h) The final output and the corresponding error.

3) *Comparison with DLMRI*: In this experiment, we compared our model with the state-of-the-art DL-based method, DLMRI, for reconstructing individual 2D cardiac MR images. The comparison was performed for 3-fold and 6-fold acceleration factors.

a) *Models*: For CNN, we selected the parameters  $n_d = 5$ ,  $n_c = 5$ . To ensure a fair comparison, we report the aggregated result on the test set from two-way cross-validation (i.e. two iterations of train on five subjects and test on the other five). For each iteration of the cross validation, the network was initialised using He initialisation and trained end-to-end. For 6-fold undersampling, we initialised the network using the parameters obtained from the trained models from 3-fold acceleration. Each network was trained until the training error

TABLE I  
DLMRI VS. CNN ACROSS 10 SCANS

	3-fold	6-fold
Models	MSE (SD) $\times 10^{-3}$	MSE (SD) $\times 10^{-3}$
DLMRI	2.12 (1.27)	6.31 (2.95)
CNN (2D)	<b>0.89 (0.46)</b>	<b>3.42 (1.65)</b>

converged, which approximately took 3 days per network on a GeForce TITAN X.

For DLMRI, we used the implementation from [2] with patch size  $6 \times 6$ . Since DLMRI is quite time consuming, in order to obtain the results within a reasonable amount of time, we trained a joint dictionary for all time frames within the subject and reconstructed them in parallel. Note that we did not observe any decrease in performance from this approach. For each subject, we ran 400 iterations and obtained the final reconstruction.

*b) Results:* The means of the reconstruction errors across 10 subjects are summarised in Table. I. For both 3-fold and 6-fold acceleration, one can see that CNN consistently outperformed DLMRI, and that the standard deviation of the error made by CNN was smaller. The reconstructions from 6-fold acceleration is in Fig. 8. Although both methods suffered from significant loss of structures, the CNN was still capable of better preserving the texture than DLMRI (highlighted in red ellipse). On the other hand, DLMRI created block-like artefacts due to over-smoothing. 6x undersampling for these images typically approaches the limit of sparsity-based methods, however, the CNN was able to predict some anatomical details which was not possible by DLMRI. This could be due to the fact that the CNNs has more free parameters to tune with, allowing the network to learn complex but more accurate end-to-end transformations of data.

*c) Comparison of Reconstruction Speed:* While training the CNN is time consuming, once it is trained, the inference can be done extremely quickly on a GPU. Reconstructing each slice took  $23 \pm 0.1$  milliseconds on a GeForce GTX 1080, which enables real-time applications. To produce the above results, DLMRI took about  $6.1 \pm 1.3$  hours per subject on CPU. Even though we do not have a GPU implementation of DLMRI, it is expected to take longer than 23ms because DLMRI requires dozens of iterations of dictionary learning and sparse coding steps. Using a fixed, pre-trained dictionary could remove this bottleneck in computation although this would likely be to the detriment of reconstruction quality.

### C. 3D Experiments

For the following experiments, we split our dataset into training and testing sets containing seven and three subjects respectively. Compared to the 2D case, we have significantly less data. In addition, working with a large input is a burden on memory, limiting the size of the network that can be used. To address this, we trained our model on an input size  $256 \times N_{patch} \times 30$ , where the direction of patch extraction corresponds to the frequency-encoding direction. In this way, we can train the network on a reduced input size while letting

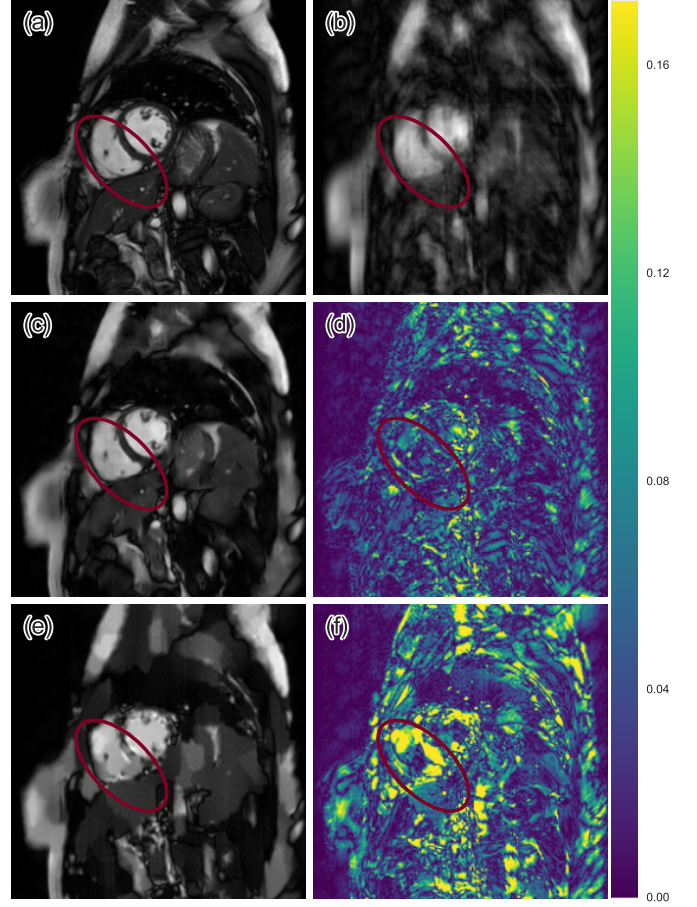


Fig. 8. The comparison of reconstructions from DLMRI and CNN. (a) The original (b) 6x undersampled (c,d) CNN reconstruction and its error map (e,f) DLMRI reconstruction and its error map. There are larger errors in (f) than (d) and red ellipse highlights the anatomy that was reconstructed by CNN better than DLMRI.

the network learn the same aliasing patterns as if the full input was used. Furthermore, we applied small elastic deformations [30] to augment the training data input in order to further increase the variation of the examples seen by the network.

*1) Effect of Data Sharing:* In this experiment, we evaluated the effect of using the features obtained from data sharing. We trained the following two networks: *D5-C10(S)* ( $n_d = 5$ ,  $n_c = 10$  with data sharing) and *D6-C10* ( $n_d = 6$ ,  $n_c = 10$  without data sharing). In the second network, the data sharing is replaced by an additional convolution layer to account for the additional input. We trained each model to reconstruct the sequences from 9-fold undersampling for  $2.5 \times 10^4$  backpropagations. Their learning is plotted in Fig. 9. We can notice that there is a considerable difference in their errors. In addition, the gap between the train and the test error of the *D5-C10(S)* is much smaller. This suggests that it was able to learn a strategy to de-alias image that generalises better. Moreover, by using data sharing, the network was able to learn faster. The visualization of their reconstructions can be found in the following section.

*2) Comparison with DLTG:* In this experiment, we compared our model with state-of-the-art, DLTG [3], for reconstructing the dynamic sequence. We compared the results for

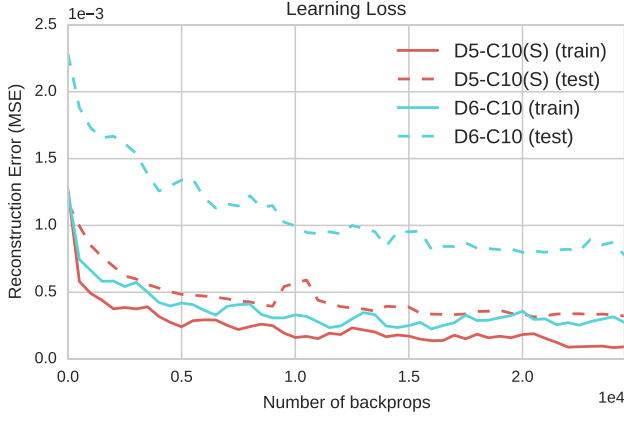


Fig. 9. The effect of data sharing. The network with data sharing shows superior performance over the other. In particular, it has considerably lower test error, showing an improved generalization property.

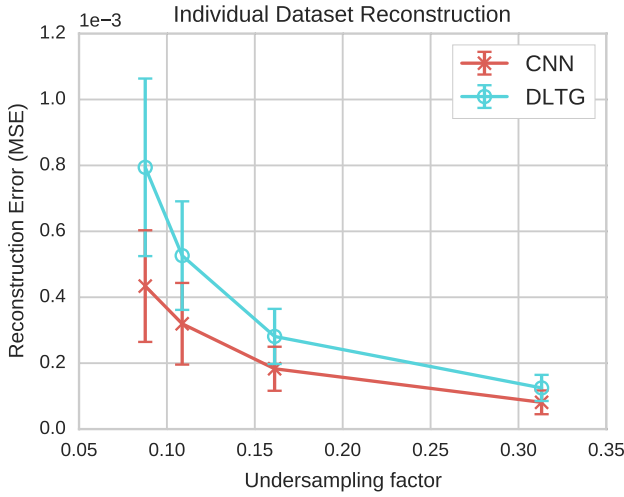


Fig. 10. The reconstruction errors of DLTG and CNN across 10 subjects for different undersampling rates.

3, 6, 9 and 11-fold acceleration factors.

*a) Models:* For the CNN, we used  $n_d = 5$ ,  $n_c = 10$  with data sharing as explained above. We also set the weight decay to 0 as we did not notice any overfitting of the model. Contrary to the 2D case, we trained each network as follows: we first pre-trained the network on various undersampling rates (0-9x) for  $5 \times 10^4$  backpropagations. Subsequently, each network was fine-tuned for a specific undersampling rate using Adam with learning rate reduced to  $5 \times 10^{-5}$  for  $10^4$  backpropagations. We performed three way cross validation (where for two iterations we train on 7 subjects then test on 3 subjects, one iteration where we train on 6 subjects and test on 4 subjects) and we aggregated the test errors. The pre-training and the fine tuning stages took approximately 3.5 days and 14 hours respectively using a GeForce GTX 1080. Since the training is time consuming, we did not train the networks longer but we speculate that the network will benefit from further training using lower learning rates. For DLTG, we used the default parameters with total variation.

*b) Result:* The final reconstruction error is summarised in Fig. 10. we see that CNN consistently outperforms DLTG

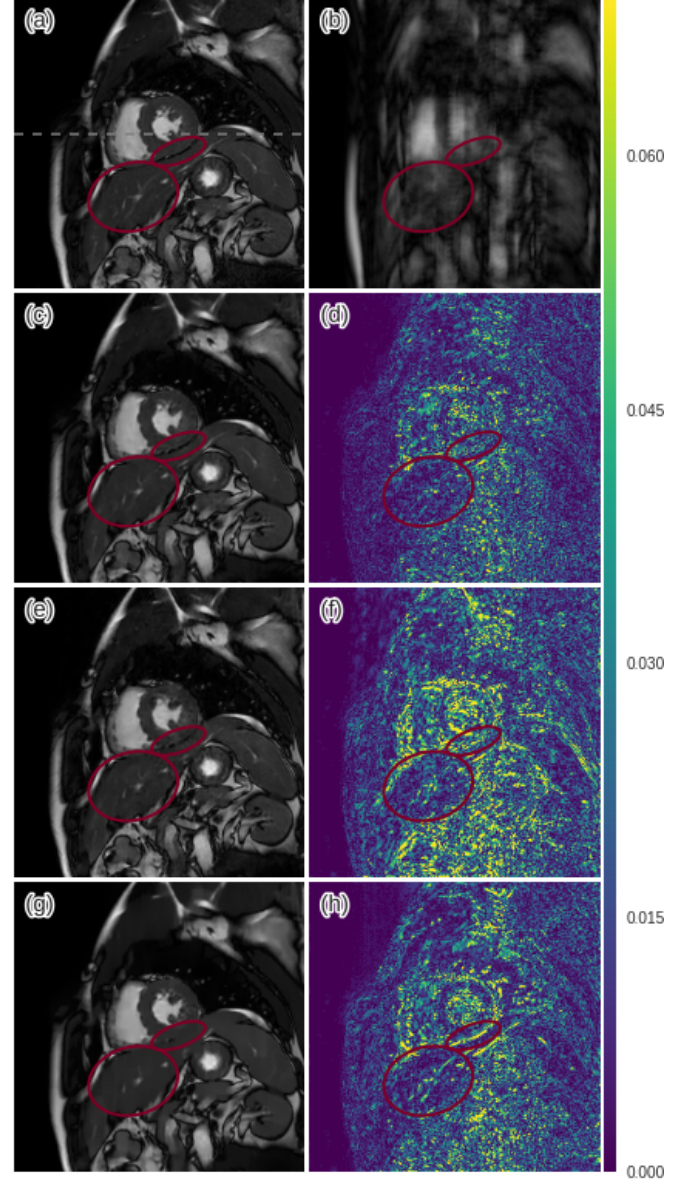


Fig. 11. The comparison of reconstructions from DLTG and CNN (a) The original (b) 9x undersampled (c,d) CNN with data sharing and its error map (e,f) CNN without data sharing and its error map (g,h) DLTG reconstruction and its error map. Red ellipses highlight the anatomy that was reconstructed by CNN better than DLTG.

for all undersampling factors. For a low acceleration factor (3x undersampling), the two methods performed approximately the same, however, for more aggressive undersampling factors, CNN was able to reduce the error by a considerable margin. The visualisation of reconstruction from 9-fold undersampling is shown in Fig. 11, including the reconstruction from the CNN without data sharing. One can see that, as with the 2D case, at aggressive undersampling rate DLTG produced blocky artefacts, whereas the CNN methods were capable of reconstructing finer details (indicated in red ellipse). On the other hand, for the CNN without data sharing, one can notice grainy noise-like artefacts. Even though it was able to reconstruct the underlying anatomy more faithfully than



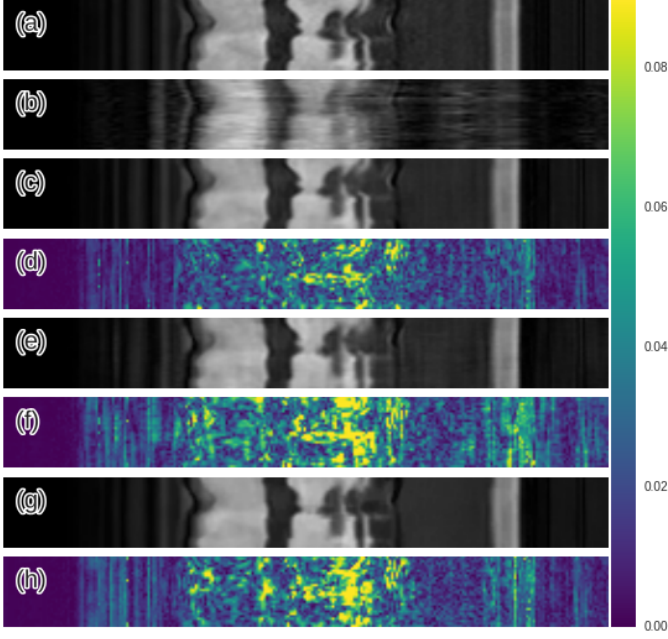


Fig. 12. The comparison of reconstructions along temporal dimension. Here we extract a 110th slice. (a) The original (b) 9x undersampled (c,d) CNN with data sharing and its error map (e,f) CNN without data sharing and its error map (g,h) DLTG reconstruction and its error map.

DLTG, the overall error was worse. However, this artefact was not present in the images reconstructed by the CNN with data sharing. Temporal profiles from the reconstructions are shown in Fig. 12. Even though the naive data sharing approach does result in data inconsistency in highly dynamic regions, the CNN was able to rectify this internally and reconstructed the correct motion with errors smaller than the other two methods. This suggests the CNN’s capability solve the joint de-aliasing and estimation of dynamic motion.

*c) Reconstruction speed:* Similar to the 2D case, the DLTG takes 6.6 hours per subject on CPU. For the CNN, each sequence was reconstructed on average  $8.21s \pm 0.02s$  on GPU GeForce GTX 1080. This is significantly slower than reconstructing 2D images as introducing a temporal axis greatly increases the computational effort of the convolution operations. Nevertheless, the reconstruction speed of our method is much faster than DLTG and is reasonably fast for offline reconstruction.

## VIII. DISCUSSION AND CONCLUSION

In this work, we evaluated the applicability of CNNs for the challenge of reconstructing undersampled cardiac MR image data. The experiments presented show that using a network with interleaved data consistency stages, it is feasible to obtain a model which can reconstruct images well. The CS framework offers a mathematical guarantee for the signal recovery, which makes the approach appealing in theory as well as in practice even though the required sparsity cannot generally be genuinely achieved in medical imaging. However, even though this is not the case for CNNs, we have empirically shown that a CNN-based approach can outperform DL-based MR

reconstruction. In addition, at very aggressive undersampling rates, the CNN method was capable of reconstructing most of the anatomical structures more accurately based on the learnt priors, while CS-based methods do not guarantee such behaviour.

It is important to note that in the experiments presented the data was produced by retrospective undersampling of back transformed complex images (equivalent to single-coil data) obtained through an original SENSE reconstruction. Although the application of CNN reconstruction needs to be investigated in the more practical scenario of full array coil data from parallel MR, the results presented show a great potential to apply deep learning for MR reconstruction. The additional richness of array coil data has the potential to further improve performance, although it will also add considerable complexity to the required CNN architecture.

In this work, we were able to show that the network can be trained using arbitrary Cartesian undersampling masks of fixed sampling rate rather than selecting a fixed number of undersampling masks for training and testing. In addition, we were able to pre-train the network on various undersampling rates before fine-tuning the network. This suggests that the network was capable of learning a generic strategy to de-alias the images. A further investigation should consider how tolerant the network is for different undersampling patterns such as radial and spiral trajectories. As these trajectories provide different properties of aliasing artefacts, a further validation is appropriate to determine the flexibility of our approach. However, radial sampling naturally fits well with the data sharing framework and therefore can be expected to push the performance of the network further. The data sharing approach may also make it feasible to adopt regular undersampling patterns which are intrinsically more efficient. Another interesting direction would be to jointly optimise the undersampling mask using the learning framework.

To conclude, although CNNs can only learn local representations which should not affect global structure, it remains to be determined how the CNN approach operates when there is a pathology present in images, or other more variable content. We have performed a cross-validation study to ensure that the network can handle unseen data acquired through the same acquisition protocol. Generalisation properties must be evaluated carefully on a larger dataset. However, CNNs are flexible in a way such that one can incorporate application specific priors to their objective functions to allocate more importance to preserving features of interest in the reconstruction, provided that such expert knowledge is available at training time. For example, analysis of cardiac images in clinical settings often employs segmentation and/or registration. Multi-task learning is a promising approach to further improve the utility of CNN-based MR reconstructions.

## ACKNOWLEDGMENT

The work was partially funded by EPSRC Programme Grant (EP/P001009/1).

## REFERENCES

- [1] M. Lustig and D. Donoho, "Compressed sensing MRI," *Signal Processing Magazine, IEEE*, vol. 25, no. 2, pp. 72–85, 2008.
- [2] S. Ravishanker and Y. Bresler, "MR Image Reconstruction From Highly Undersampled k-Space Data by Dictionary Learning," *IEEE Transactions on Medical Imaging*, vol. 30, pp. 1028–1041, may 2011.
- [3] J. Caballero, A. N. Price, D. Rueckert, and J. V. Hajnal, "Dictionary Learning and Time Sparsity for Dynamic MR Data Reconstruction," *{IEEE} Trans Med Imaging*, vol. 33, no. 4, pp. 979–994, 2014.
- [4] H. Jung, J. C. Ye, and E. Y. Kim, "Improved k t BLAST and k t SENSE using FOCUS," *Magnetic Resonance in Medicine*, vol. 52, pp. 3201–3226, 2007.
- [5] T. M. Quan and W.-k. Jeong, "Compressed sensing reconstruction of dynamic contrast enhanced MRI using GPU-accelerated convolutional sparse coding," *2016 IEEE 13th International Symposium on Biomedical Imaging (ISBI)*, pp. 518–521, 2016.
- [6] A. Hirabayashi, N. Inamuro, K. Mimura, T. Kurihara, and T. Homma, "Compressed sensing MRI using sparsity induced from adjacent slice similarity," in *2015 International Conference on Sampling Theory and Applications (SampTA)*, pp. 287–291, IEEE, may 2015.
- [7] U. Martin, S. S. Vasanawala, and M. Lustig, "ESPIRiT: An Eigenvalue Approach to Autocalibrating Parallel MRI: Where SENSE meets GRAPPA," *Magnetic Resonance in Medicine*, vol. 71, no. 3, pp. 990–1001, 2014.
- [8] J. Huang, C. Chen, and L. Axel, "Fast multi-contrast MRI reconstruction," *Magnetic Resonance Imaging*, vol. 32, no. 10, pp. 1344–1352, 2014.
- [9] X. Peng and D. Liang, "MR Image Reconstruction with Convolutional Characteristic Constraint (CoCCO)," *IEEE SIGNAL PROCESSING LETTERS*, vol. 22, no. 8, pp. 1184–1188, 2015.
- [10] K. H. Jin, D. Lee, and J. C. Ye, "A novel k-space annihilating filter method for unification between compressed sensing and parallel MRI," *Proceedings - International Symposium on Biomedical Imaging*, vol. 2015-July, pp. 327–330, 2015.
- [11] D. Liang, B. Liu, J. Wang, and L. Ying, "Accelerating SENSE using compressed sensing," *Magnetic Resonance in Medicine*, vol. 62, pp. 1574–1584, dec 2009.
- [12] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," *Arxiv.Org*, vol. 7, no. 3, pp. 171–180, 2015.
- [13] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," *Nips*, pp. 1–10, 2015.
- [14] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, pp. 234–241, may 2015.
- [15] C. Dong, C. C. Loy, K. He, and X. Tang, "Image Super-Resolution Using Deep Convolutional Networks," *IEEE transactions on pattern analysis and machine intelligence*, vol. 38, pp. 295–307, feb 2016.
- [16] W. Shi, J. Caballero, F. Huszar, J. Totz, A. P. Aitken, R. Bishop, D. Rueckert, and Z. Wang, "Real-Time Single Image and Video Super-Resolution Using an Efficient Sub-Pixel Convolutional Neural Network," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, jun 2016.
- [17] A. Kulkarni, Kuldeep and Lohit, Suhas and Turaga, Pavan and Kerviche, Ronan and Ashok, "ReconNet: Non-Iterative Reconstruction of Images From Compressively Sensed Measurements," *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.
- [18] Y. Yang, J. Sun, H. Li, and Z. Xu, "Deep ADMM-Net for Compressive Sensing MRI," in *Advances in Neural Information Processing Systems 29* (D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, eds.), pp. 10–18, Curran Associates, Inc., 2016.
- [19] W. Shanshan, S. Zhenghang, Y. Leslie, P. Xi, Z. Shun, L. Feng, F. Dagan, and D. Liang, "Accelerating magnetic resonance imaging via deep learning," in *IEEE 13th International Symposium on Biomedical Imaging (ISBI)*, pp. 514–517, 2016.
- [20] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," in *Nature*, 323, pp. 533–536, 1986.
- [21] A. Faijul and K. Murase, *Learning algorithms in Complex-Valued Neural Networks Using Wirtinger Calculus*. The Institute of Electrical and Electronics Engineers, Inc, 2013.
- [22] S. J. Riederer, T. Tasciyan, F. Farzaneh, J. N. Lee, R. C. Wright, and R. J. Herfkens, "Mr fluoroscopy: Technical feasibility," *Magnetic resonance in medicine*, vol. 8, no. 1, pp. 1–15, 1988.
- [23] V. Rasche, R. W. D. Boer, D. Holz, and R. Proksa, "Continuous radial data acquisition for dynamic mri," *Magnetic Resonance in Medicine*, vol. 34, no. 5, pp. 754–761, 1995.
- [24] S. Zhang, K. T. Block, and J. Frahm, "Magnetic resonance imaging in real time: Advances using radial flash," *Journal of Magnetic Resonance Imaging*, vol. 31, no. 1, pp. 101–109, 2010.
- [25] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going Deeper With Convolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–9, 2015.
- [26] F. Yu and V. Koltun, "Multi-Scale Context Aggregation by Dilated Convolutions," *Iclr*, pp. 1–9, 2016.
- [27] K. He, X. Zhang, S. Ren, and J. Sun, "Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification," *arXiv preprint*, pp. 1–11, 2015.
- [28] D. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," *arXiv:1412.6980 [cs]*, pp. 1–15, 2014.
- [29] K. P. Pruessmann, M. Weiger, M. B. Scheidegger, and P. Boesiger, "SENSE: Sensitivity encoding for fast MRI," *Magnetic Resonance in Medicine*, vol. 42, no. 5, pp. 952–962, 1999.
- [30] P. Y. Simard, D. Steinkraus, J. C. Platt, *et al.*, "Best practices for convolutional neural networks applied to visual document analysis," in *ICDAR*, vol. 3, pp. 958–962, Citeseer, 2003.